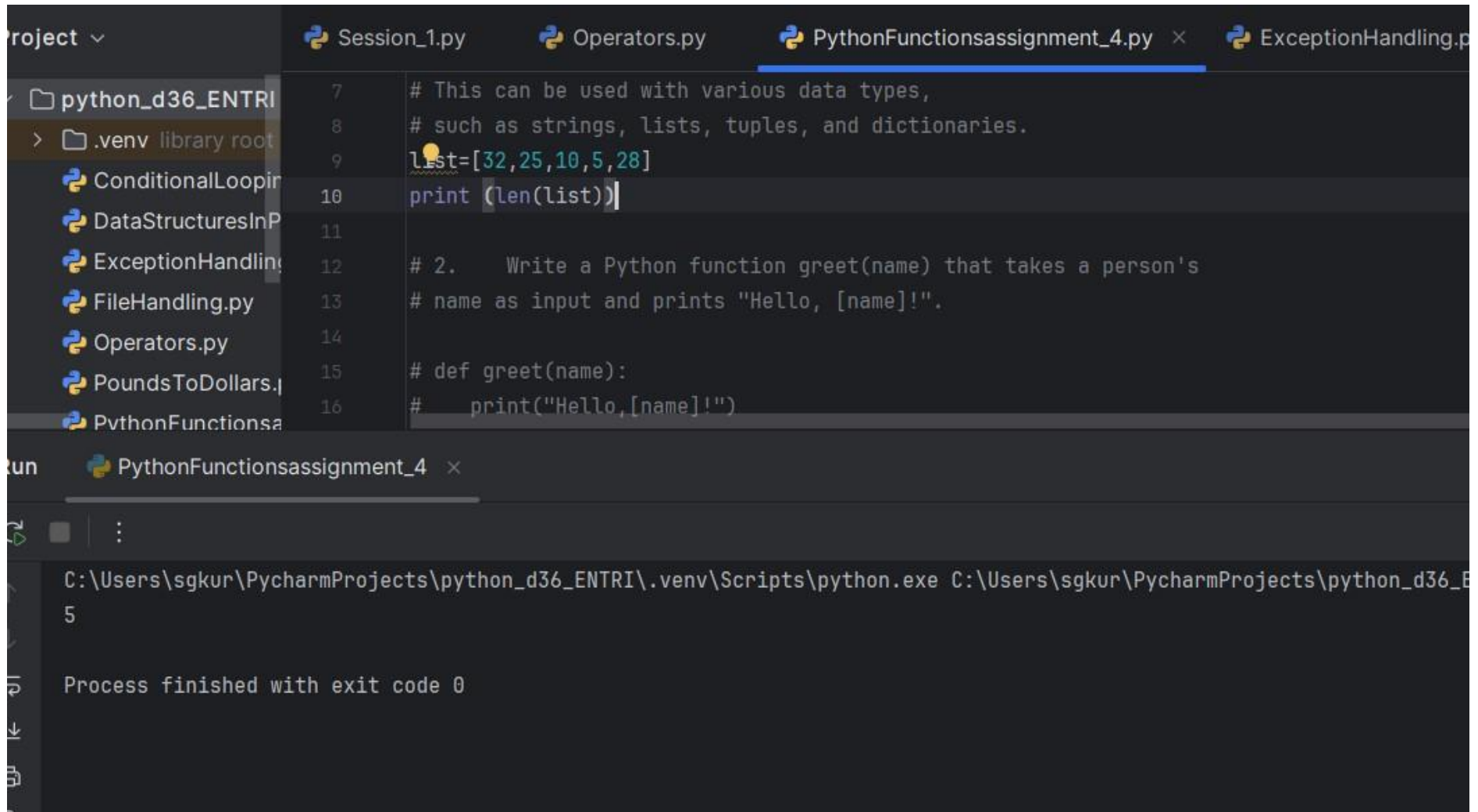


ASSIGNMENT ON FUNCTIONS IN PYTHON

By
Shino Mary Philipose

Q.1



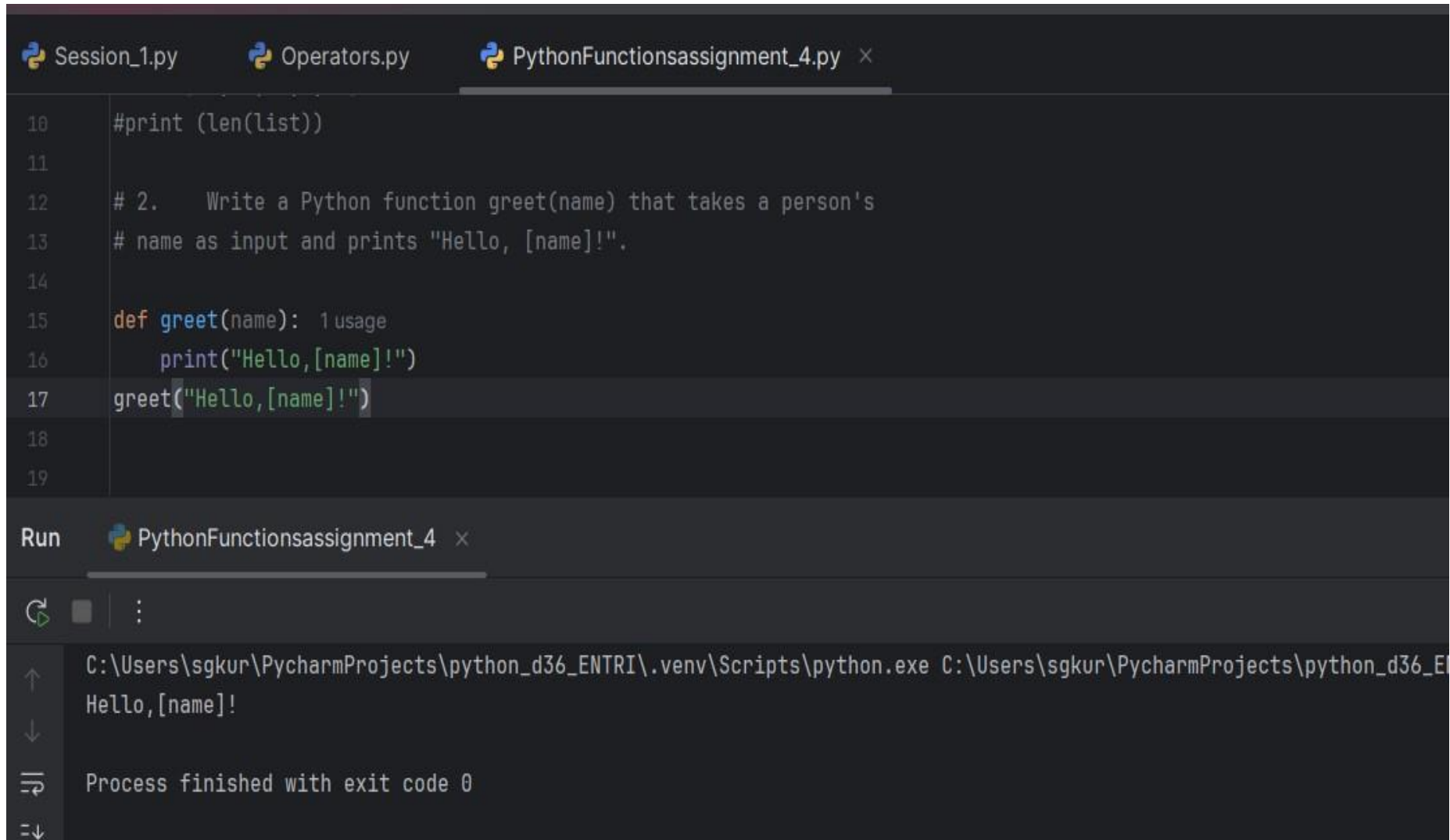
The screenshot shows the PyCharm IDE interface. The top toolbar contains icons for Session_1.py, Operators.py, PythonFunctionsassignment_4.py (which is the active file), and ExceptionHandling.p. The left sidebar displays a project tree for 'python_d36_ENTRI' with a subfolder '.venv library root' containing files like ConditionalLoopin, DataStructuresInP, ExceptionHandling, FileHandling.py, Operators.py, PoundsToDollars.i, and PvthonFunctionsa. The main editor window shows the code for PythonFunctionsassignment_4.py with line numbers 7 through 16. The code includes comments about data types, a list definition, a print statement for the list length, and a function definition for 'greet'. The bottom panel shows the 'Run' output for 'PythonFunctionsassignment_4', displaying the command path and the output '5', followed by the message 'Process finished with exit code 0'.

```
7 # This can be used with various data types,  
8 # such as strings, lists, tuples, and dictionaries.  
9 list=[32,25,10,5,28]  
10 print (len(list))  
11  
12 # 2. Write a Python function greet(name) that takes a person's  
13 # name as input and prints "Hello, [name]!".  
14  
15 # def greet(name):  
16 #     print("Hello,[name]!")
```

Run PythonFunctionsassignment_4 ×

C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\python_d36_ENTRI.py
5
Process finished with exit code 0

Q.2



The image shows a PyCharm IDE window with three tabs: 'Session_1.py', 'Operators.py', and 'PythonFunctionsassignment_4.py'. The 'PythonFunctionsassignment_4.py' tab is active, displaying a Python script. The script includes a comment for a function 'greet' and a call to the function. Below the code editor, the 'Run' toolbar is visible, and the 'Run' output window shows the execution path and the output 'Hello, [name]!'. The process finished with exit code 0.

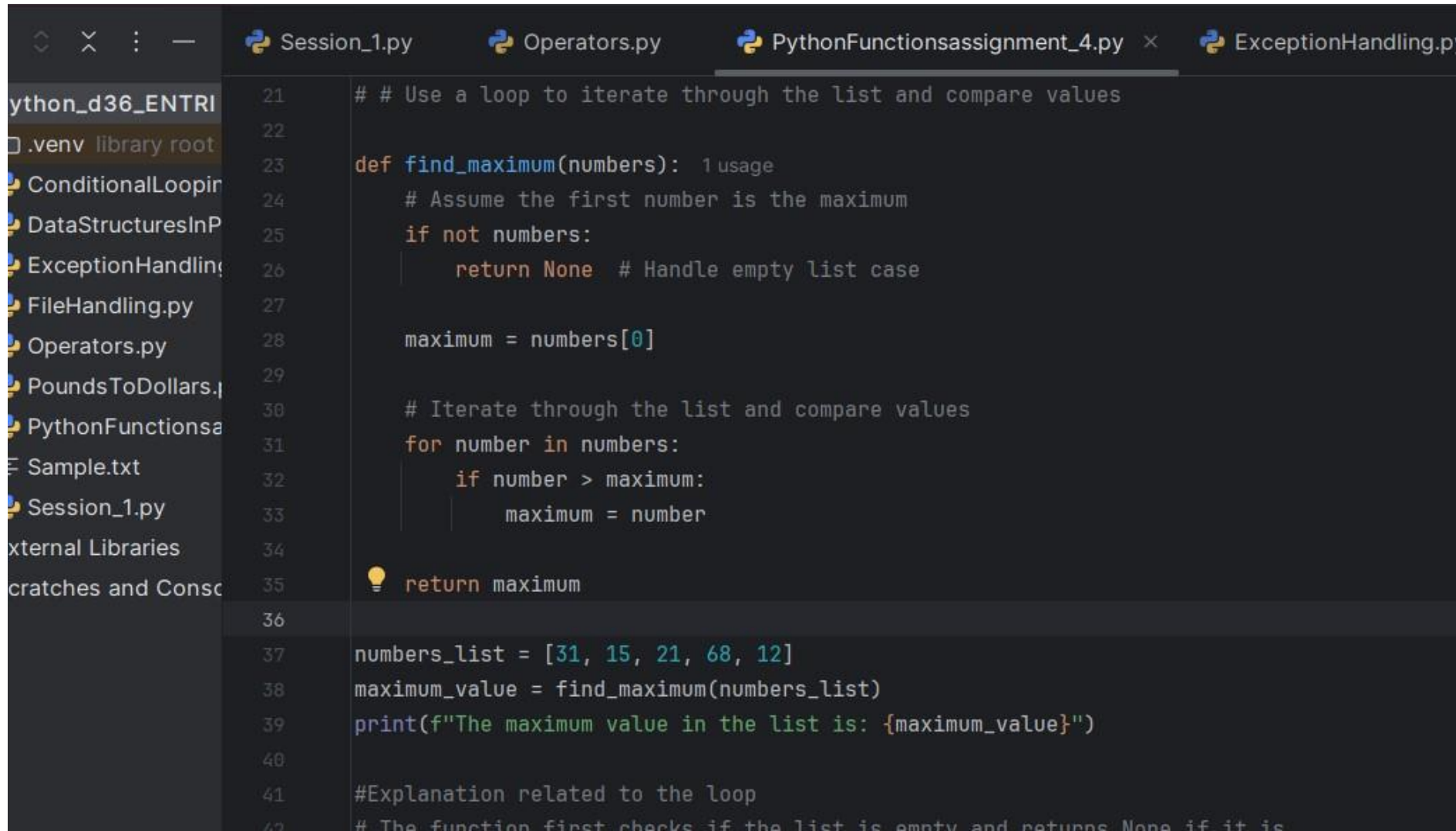
```
10 #print (len(list))
11
12 # 2. Write a Python function greet(name) that takes a person's
13 # name as input and prints "Hello, [name]!".
14
15 def greet(name): 1 usage
16     print("Hello,[name]!")
17 greet("Hello,[name]!")
18
19
```

Run PythonFunctionsassignment_4

C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\python_d36_ENTRI.py
Hello, [name]!

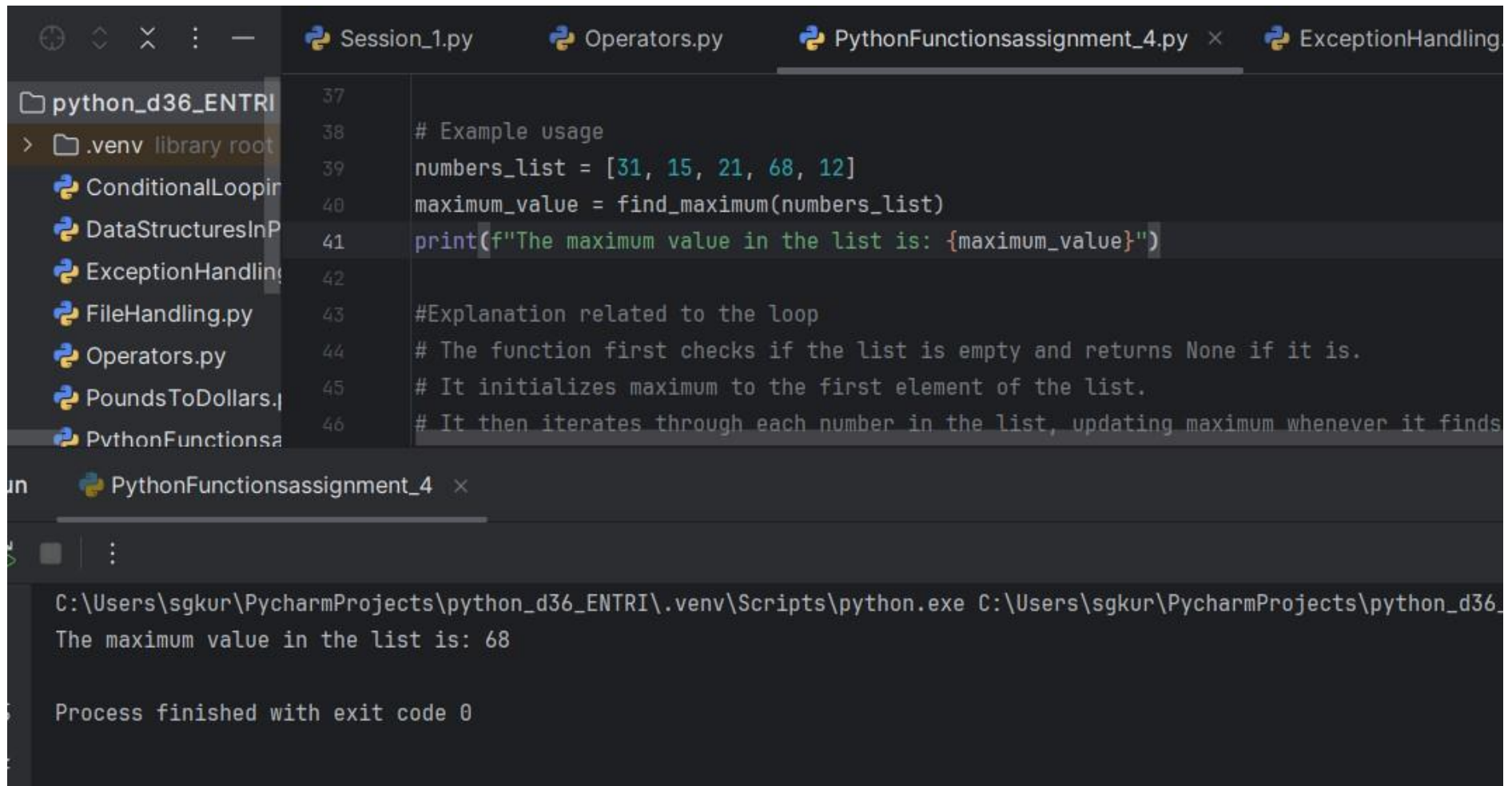
Process finished with exit code 0

Q.3



```
21  ## Use a loop to iterate through the list and compare values
22
23  def find_maximum(numbers): 1 usage
24      # Assume the first number is the maximum
25      if not numbers:
26          return None # Handle empty list case
27
28      maximum = numbers[0]
29
30      # Iterate through the list and compare values
31      for number in numbers:
32          if number > maximum:
33              maximum = number
34
35      💡 return maximum
36
37  numbers_list = [31, 15, 21, 68, 12]
38  maximum_value = find_maximum(numbers_list)
39  print(f"The maximum value in the list is: {maximum_value}")
40
41  #Explanation related to the loop
42  # The function first checks if the list is empty and returns None if it is
```

Q.3



The image shows a PyCharm IDE window with several tabs open: Session_1.py, Operators.py, PythonFunctionsassignment_4.py, and ExceptionHandling.py. The active tab is PythonFunctionsassignment_4.py, which contains the following code:

```
37
38 # Example usage
39 numbers_list = [31, 15, 21, 68, 12]
40 maximum_value = find_maximum(numbers_list)
41 print(f"The maximum value in the list is: {maximum_value}")
42
43 #Explanation related to the loop
44 # The function first checks if the list is empty and returns None if it is.
45 # It initializes maximum to the first element of the list.
46 # It then iterates through each number in the list, updating maximum whenever it finds
```

The output console at the bottom shows the command executed and the result:

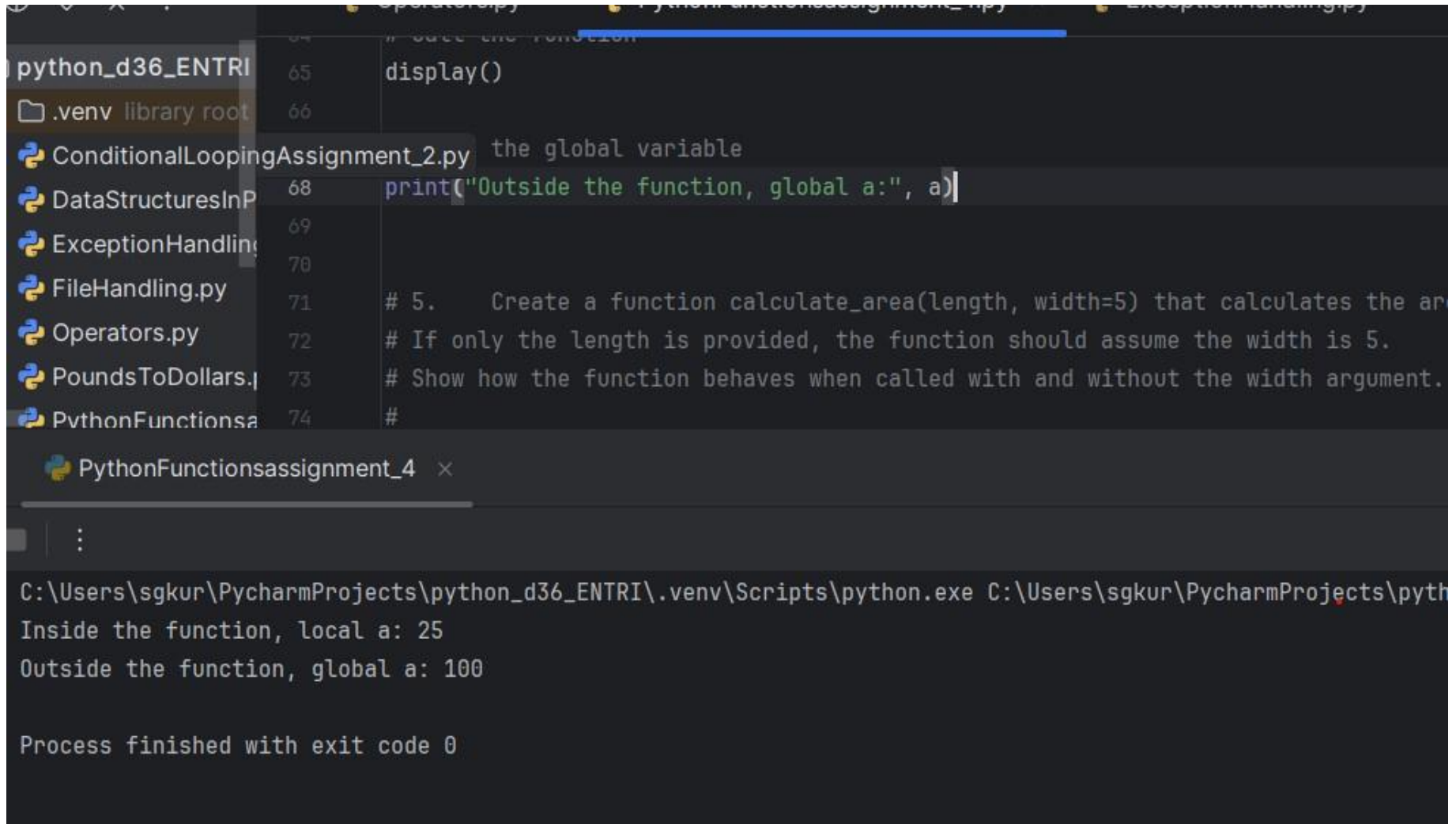
```
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_
The maximum value in the list is: 68

Process finished with exit code 0
```

Q.4

```
Python_d36_ENTRI 51 # Write a program where a global variable and a local variable have the same name and
                  52 # show how Python differentiates between them.
                  53 #💡 Python ,variables have a scope that determines where they can be accessed from.
                  54 #A variable's scope is the region of the program where it is defined and can be used.
                  55 #
                  56 # Global variable
                  57 a= 100
                  58
                  59 def display(): 1 usage
                  60     # Local variable with the same name
                  61     a = 25
                  62     print("Inside the function, local a:", a)
                  63
                  64 # Call the function
                  65 display()
                  66
                  67 # Print the global variable
                  68 print("Outside the function, global a:", a)
                  69
                  70
```


Q.4



The screenshot shows a PyCharm IDE with a project named 'python_d36_ENTRI'. The file explorer on the left lists several files: '.venv library root', 'ConditionalLoopingAssignment_2.py', 'DataStructuresInP...', 'ExceptionHandling...', 'FileHandling.py', 'Operators.py', 'PoundsToDollars.py', and 'PvthonFunctionsa...'. The main editor window displays a Python script with the following content:

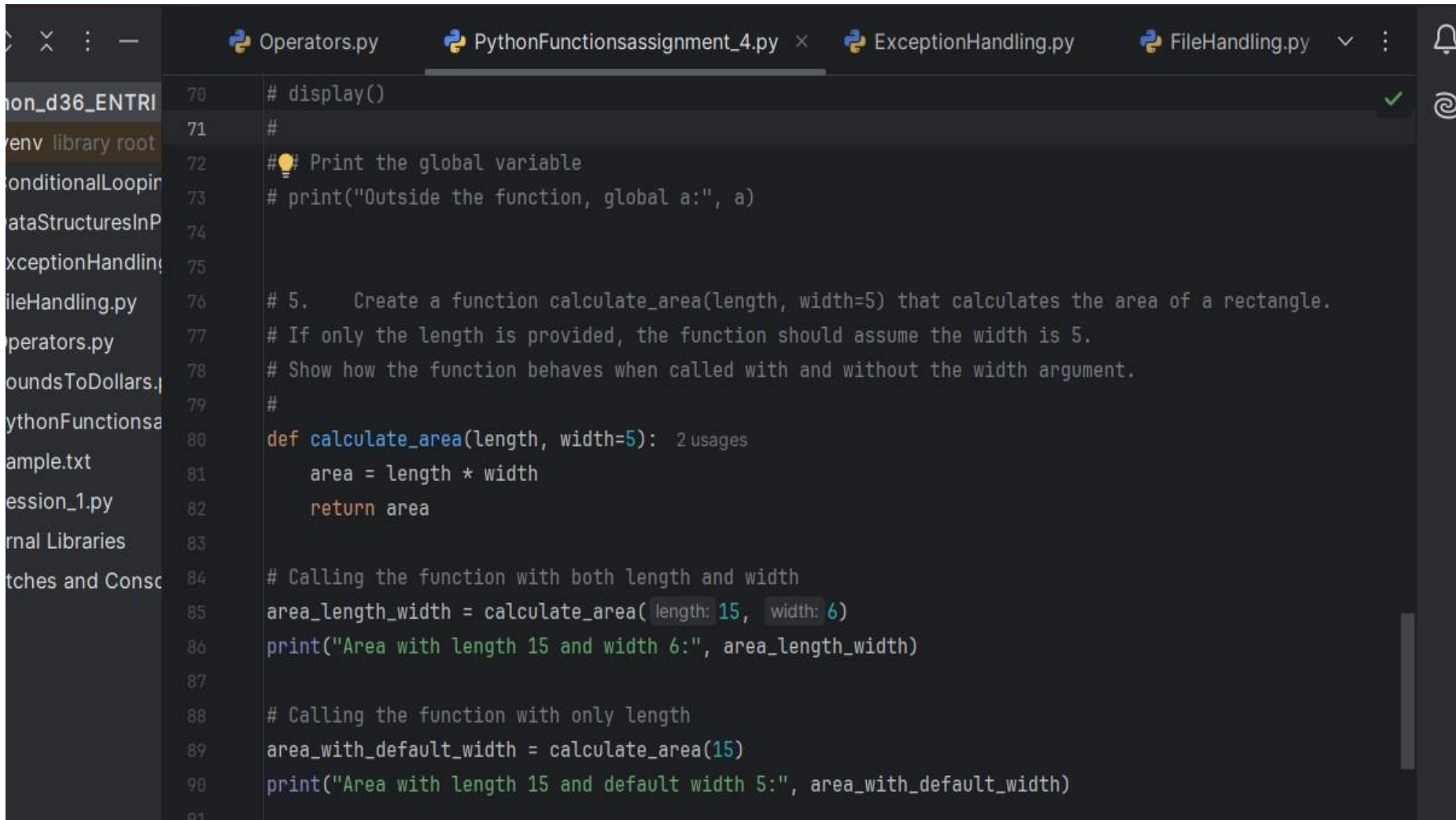
```
64 # Save the function
65 display()
66
67 # the global variable
68 print("Outside the function, global a:", a)
69
70
71 # 5. Create a function calculate_area(length, width=5) that calculates the ar
72 # If only the length is provided, the function should assume the width is 5.
73 # Show how the function behaves when called with and without the width argument.
74 #
```

Below the editor, a terminal window titled 'PythonFunctionsassignment_4' shows the output of the script:

```
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\pyth
Inside the function, local a: 25
Outside the function, global a: 100

Process finished with exit code 0
```

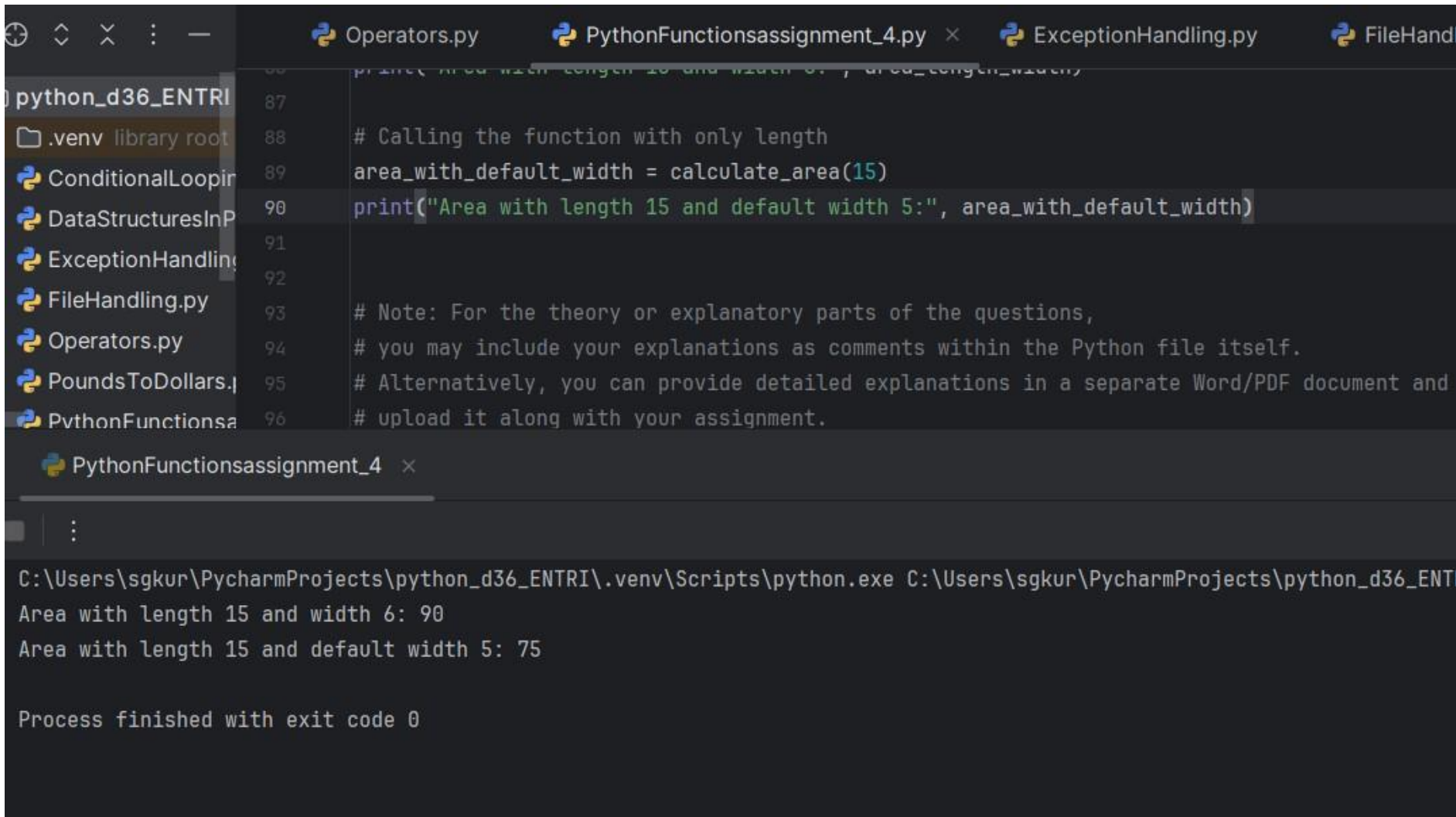
Q.5



The screenshot shows a code editor with a dark theme. The top bar displays four open files: Operators.py, PythonFunctionsassignment_4.py (which is the active file), ExceptionHandling.py, and FileHandling.py. The left sidebar shows a file explorer with various files and folders. The main editor area displays the following Python code:

```
70 # display()
71 #
72 #💡 Print the global variable
73 # print("Outside the function, global a:", a)
74
75
76 # 5. Create a function calculate_area(length, width=5) that calculates the area of a rectangle.
77 # If only the length is provided, the function should assume the width is 5.
78 # Show how the function behaves when called with and without the width argument.
79 #
80 def calculate_area(length, width=5): 2 usages
81     area = length * width
82     return area
83
84 # Calling the function with both length and width
85 area_length_width = calculate_area(length: 15, width: 6)
86 print("Area with length 15 and width 6:", area_length_width)
87
88 # Calling the function with only length
89 area_with_default_width = calculate_area(15)
90 print("Area with length 15 and default width 5:", area_with_default_width)
91
```


Q.5



```
python_d36_ENTRI
.venv library root
ConditionalLoopin
DataStructuresInP
ExceptionHandling
FileHandling.py
Operators.py
PoundsToDollars.i
PythonFunctionsa

Operators.py PythonFunctionsassignment_4.py x ExceptionHandling.py FileHandl

86 print("Area with length 15 and width 6:", area_length_width)
87
88 # Calling the function with only length
89 area_with_default_width = calculate_area(15)
90 print("Area with length 15 and default width 5:", area_with_default_width)
91
92
93 # Note: For the theory or explanatory parts of the questions,
94 # you may include your explanations as comments within the Python file itself.
95 # Alternatively, you can provide detailed explanations in a separate Word/PDF document and
96 # upload it along with your assignment.

PythonFunctionsassignment_4 x

C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENT
Area with length 15 and width 6: 90
Area with length 15 and default width 5: 75

Process finished with exit code 0
```

Explanations for the functions used

1. What does the len() function do in Python?

The len() function in Python returns the number of items in an object.

This can be used with various data types,
such as strings, lists, tuples, and dictionaries.

Write a Python function find_maximum(numbers) that takes a
list of integers and returns the maximum value without using the built-in max()
function.

Use a loop to iterate through the list and compare values

Explanation related to the loop

The function first checks if the list is empty and returns None if it is.

It initializes maximum to the first element of the list.

It then iterates through each number in the list, updating maximum whenever it finds a larger value.

Finally, it returns the maximum value found.

Explanations for the functions used

4. Explain the difference between local and global variables in a Python function.
Write a program where a global variable and a local variable have the same name and
show how Python differentiates between them.

In Python ,variables have a scope that determines where they can be accessed from.
#A variable's scope is the region of the program where it is defined and can be used.
#The variable created above the user defined function is considered as the global variable,
#whereas the variable created inside or within the function is considered as local variable.
#The print command inside the function will return the local variable.
#The print command after the function call will return the Global variable.

Explanations for the functions used

```
5.Create a function calculate_area(length, width=5) that calculates the area of a
rectangle.
# If only the length is provided, the function should assume the width is 5.
# Show how the function behaves when called with and without the width argument.
#
For calculating the area length and width are required.
#In the user defined function calculate_area length and width are to be given while
# defining the function.Here width =5 is given as default argument.
#While calling the function with both length and width,both arguments are given and the
#area is printed.
#If only the value for length is given then the value for the width will be taken by the
#interpreter is the default argument.Thus, the area is calculated in the second case.
```